Supporting Information

Machine Vision-Enabled Surface Temperature Mapping Based on Thermo-Responsive Cholesteric Liquid Crystal Elastomer Arrays

Haotian Zhao[a,b], Jiaqi Cheng[a,b], Jianji Wang[c], Shu Xiao[a,b], Nour F. Attia[d], Mingzhu Liu[e,f,*], Saihua Jiang[a,b,*]

[a]Institute of Safety Science and Engineering, School of Mechanical and Automotive Engineering, South China University of Technology, Wushan Road 381, Guangzhou, 510641, China

[b]Guangdong Provincial Key Laboratory of Technique and Equipment for Macromolecular Advanced Manufacturing, South China University of Technology, Guangzhou, 510641, China

[c]School of Mechanical Engineering, Long Dong University, Qingyang 745000, China.

[d]Chemistry Division, National Institute of Standards (NIS), El-Sadat (Tersa) St., El Haram, Giza, Egypt, P.O. Box 136 Giza- Code 12211,Giza, Egypt

[e]Key Laboratory of Bioinspired Smart Interfacial Science and Technology of Ministry of Education, School of Chemistry, Beihang University,Beijing 100191, China

[f]Center for Bioinspired Science and Technology, Hangzhou International Innovation Institute, Beihang University, Hangzhou 311115, China

Corresponding Author: Saihua Jiang
E-mail: liumingzhu@buaa.edu.cn; meshjiang@scut.edu.cn;

Contents

Note S1. Derivation of the relationship between wavelength and length change

The initial length, width, and thickness are set as L_0, b_0, and h_0, and the values after phase transition deformation are set as L, b, and h, respectively. Assuming the total volume remains constant during deformation and the strain rates in directions perpendicular to L-direction are the same, the volume (V) relation can be listed as:

$$V = L_0 b_0 h_0 = Lbh \tag{1}$$

From (1), by equivalent transformation, we can get (2):

$$\frac{L}{L_0} \times L_0 \times \sqrt{\frac{L_0}{L}} \times b_0 \times \sqrt{\frac{L_0}{L}} \times h_0 = Lbh = V \tag{2}$$

Since the strain rates in the width and thickness directions are the same, we also have (3):

$$h = \sqrt{\frac{L_0}{L}} h_0 \tag{3}$$

Assuming the number of helical pitches in the helical structure of the cholesteric liquid crystal is m, and the pitch length is p, we can have:

$$h = mp \tag{4}$$

According to Bragg's law of reflection:

$$\lambda = np \tag{5}$$

combining the (4) and (5), we can get (6):

$$h = \frac{m}{n} \lambda \tag{6}$$

If the center wavelength of the reflected light before film deformation is λ_0, we similarly have (7):

$$h_0 = \frac{m}{n} \lambda_0 \tag{7}$$

Because m and n are related to the intrinsic properties of the material and remain unchanged. Combining the (3), (6) and (7), we obtain (8):

$$\lambda = \sqrt{\frac{L_0}{L}} \lambda_0 \tag{8}$$

Equation (8) can be used to assist in verifying the rationality of the curve fitting in.

Note S2. Color-based Temperature Value Prediction Model (CTVPM)

The input images are converted to RGB format with specified dimensions. A large amount of training data are thoroughly shuffled and then packaged into fixed-size batches before being fed into the model for training. The training process employed in the models involves several key steps, each serving a specific purpose in transforming input RGB images into accurate temperature predictions.

(1) Convolutional Layers (Conv2D + BN + ReLU):

The process begins with convolutional layers (Conv2D) that apply filters to the input image, extracting essential features. In this model, the color features are extracted by convolutional layers. Batch normalization (BN) is applied to stabilize and accelerate training, while the ReLU (Rectified Linear Unit) activation function introduces non-linearity, enabling the model to capture complex patterns.

(2) Pooling Layers (MaxPool):

Pooling layers reduce the spatial dimensions of the feature maps. MaxPooling selects the maximum value from a specific region, effectively downsampling the image while retaining the most prominent features. This step aids in reducing the computational load and the risk of overfitting.

(3) Dropout Layers:

Dropout layers are introduced to prevent overfitting by randomly setting a fraction of the input units to zero during training. This forces the model to learn more robust features that are not reliant on any single neuron and enhance model’s generalization ability.

(4) Fully Connected Layers (Dense + ReLU):

Fully connected layers combine these features to form a comprehensive understanding of the input data. The ReLU activation function is again used to introduce non-linearity, enhancing the model's ability to make accurate predictions.

(5) Output Layer:

The final dense layer produces the output, which in this model is the predicted temperature. The output temperature values are compared with the corresponding true values, and the loss is calculated using a loss function. The loss function used in CTVPM is Mean Squared Error (MSE),

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

where y_i is the true temperature value, $y^\wedge_i$ is the temperature prediction value output by the model, and n is the number of samples. This also marks the completion of one training epoch. In the next training epoch, the model training parameters are adjusted and optimized based on the results of the current epoch.

After all training epochs are completed, the resulting model training data files are available for subsequent temperature prediction.

Note S3. Color Array-based Temperature Mapping Model (CATMM)

Color Array-based Temperature Mapping Model uses a Generative Adversarial Network (GAN). Unlike CTVPM, its output is not a numerical value but an image. The GAN is composed of two main models: a Generator and a Discriminator, which work together in an adversarial manner to improve the accuracy of temperature prediction from the input images.

The input images are converted to RGB format with specified dimensions. And the pixel values are normalized to the range [-1, 1]. This normalization helps stabilize the training process by ensuring consistent input data across different images. Then the preprocessed RGB images are put into Generator and Discriminator.

(1) Generator Model:

The Generator starts with a series of convolutional layers that reduce the spatial dimensions of the input image while extracting important features. This is followed by transposed convolutional layers (often referred to as "deconvolutional layers") that upsample the feature maps back to the original image dimensions. The final layer uses a tanh activation function to output the generated infrared image, which is then reshaped to the desired output dimensions. The image generated by the Generator is referred to as “fake image” or “output image”.

(2) Discriminator Model:

The Discriminator follows a similar structure as the generator's encoder, but the convolutional layers in Discriminatore will extract features from “fake image” and temperature infrard image (also referred to “real image”) simultaneously. The Discriminator ends with a fully connected layer, where a sigmoid activation function compares the previously extracted features from the two types of images and ultimately outputs the probability that the "fake image" is real.

(3) Loss Functions:

The training process uses binary cross-entropy loss as the primary loss function for both the generator and discriminator. The Discriminator's loss is computed by comparing its predictions for real images against true labels (ones) and its predictions for fake images against false labels (zeros). The total loss is the sum of these two components. The Generator's loss is calculated based on the discriminator's ability to correctly identify the generated images as fake. The generator is optimized to minimize this loss, effectively "fooling" the discriminator into classifying fake images as real. Both loss functions are normalized by the global batch size to ensure consistent scaling across different training setups.

Overall, for each training epoch, a batch of fake images is generated by the Generator based on the input image features. These fake images are then input into the Discriminator along with the real images, where the discriminator determines which is real and which is fake based on the extracted features. The weights are updated using the Adam optimizer according to the results, and then a new training epoch begins. This process is repeated, enabling the generator to produce images that closely resemble real images. After all training epochs, both the generator and discriminator models are saved for future use.

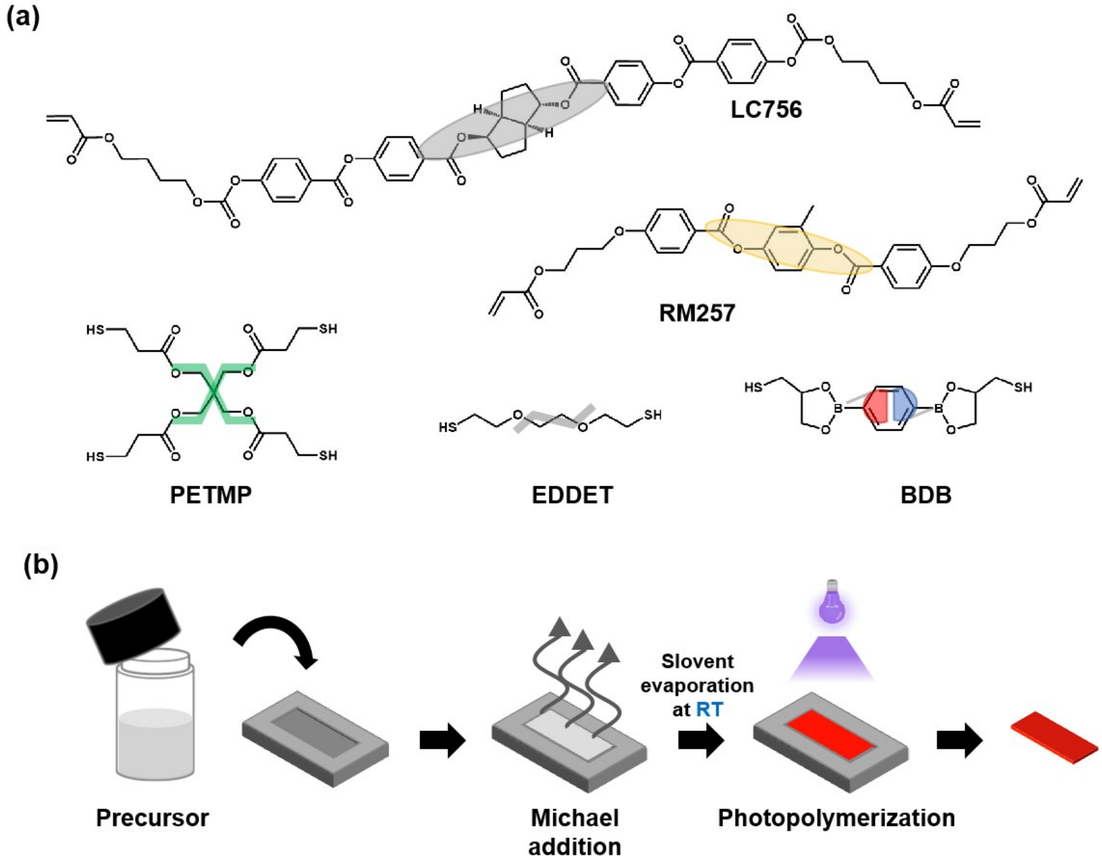


Figure S1. (a) Molecular structures of materials used for fabricating BDB-containing cholesteric liquid crystal elastomer (B-CLCE). (b) Schematic diagram of the preparation process of red-reflective B-CLCE. RT: Room Temperature.

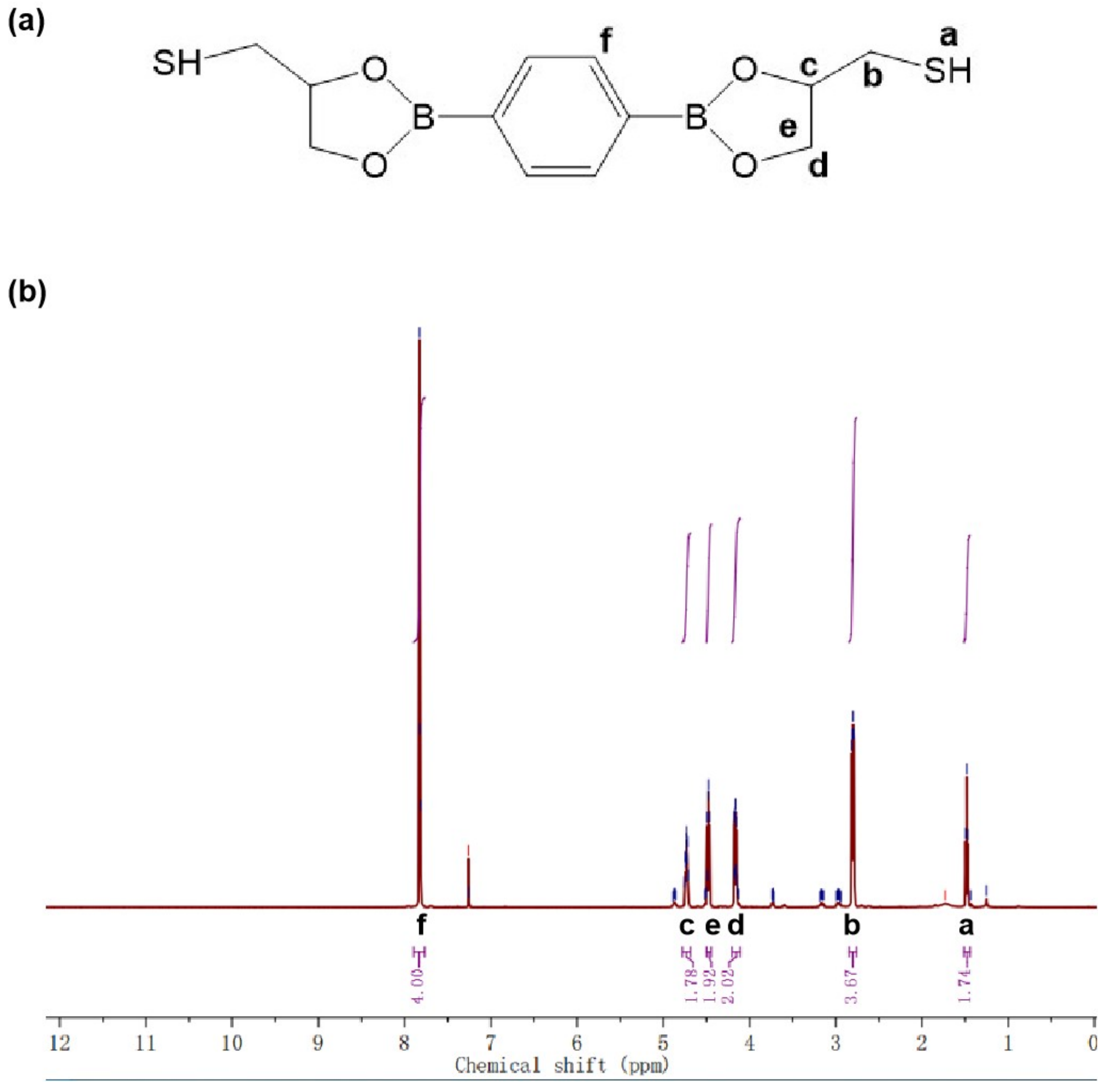


Figure S2. (a) The chemical structure and (b) ^{1}H NMR spectrum of BDB. ^{1}H NMR ($CDCl_3$, 500 MHz): BDB-δ 7.83 (s, 4H), 4.74 (m, 2H),4.49 (dd, J = 8 Hz, 7 Hz, 2H), 4.18 (dd, J = 13 Hz, 5.5 Hz, 2H), 2.82 (dd, J = 7.5 Hz, 5 Hz, 4H), 1.49 (t, J = 7.5Hz, 2H)

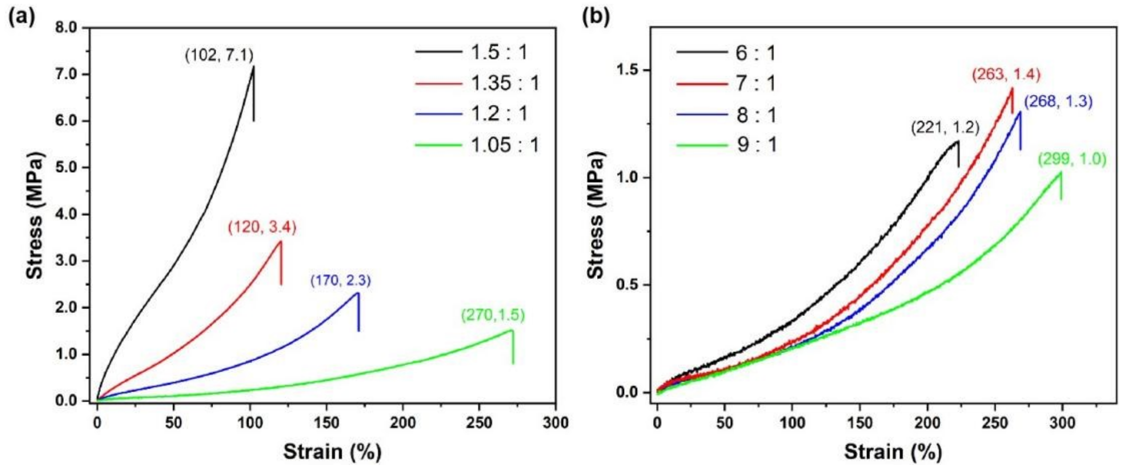


Figure S3. Stress-strain curves of B-CLCE at different molar ratios of (a) acrylates to thiols and (b) dithiols to tetrathiols.

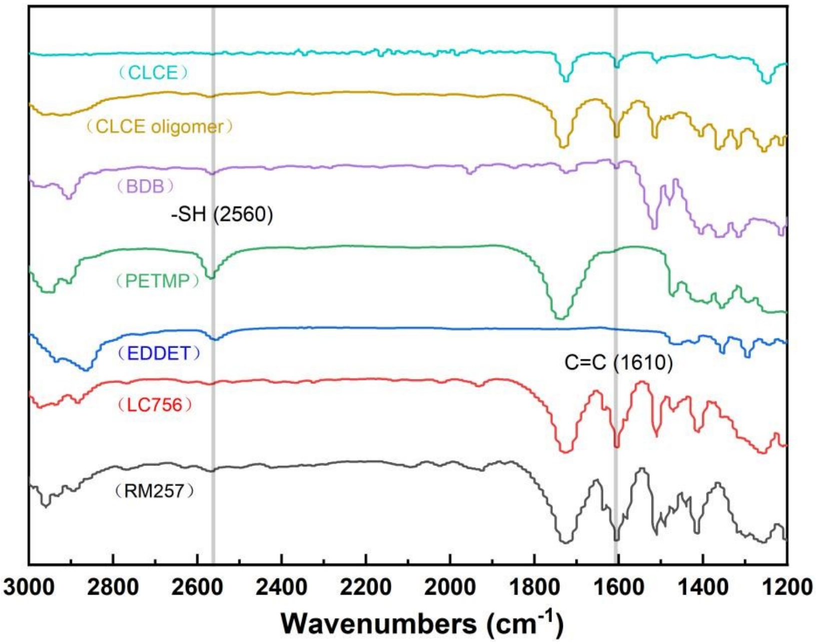


Figure S4. The FTIR spectra of RM257, LC756, EDDET, PETMP, BDB, B-CLCE oligomer and B-CLCE.

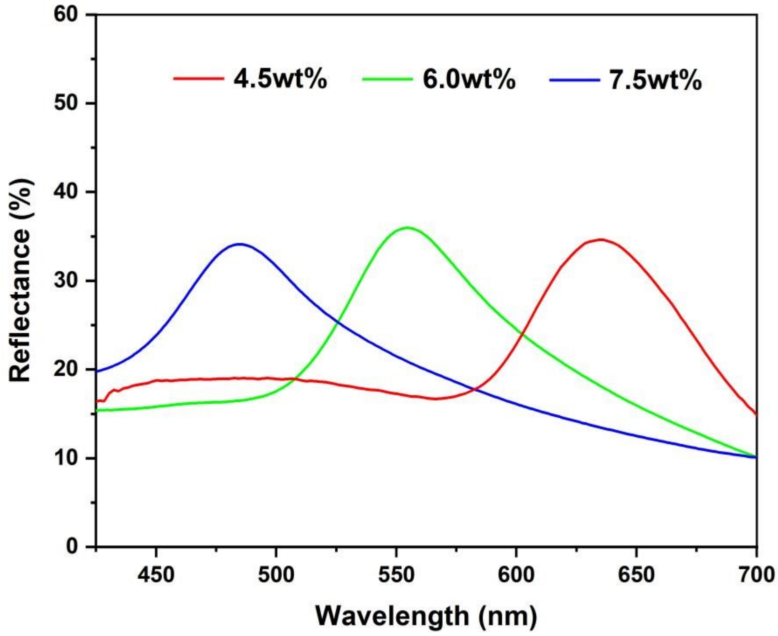


Figure S5. UV-vis reflection spectra of B-CLCE at room temperature when the mass fractions of LC756 are 4.5 wt%, 6.0 wt%, and 7.5 wt%.

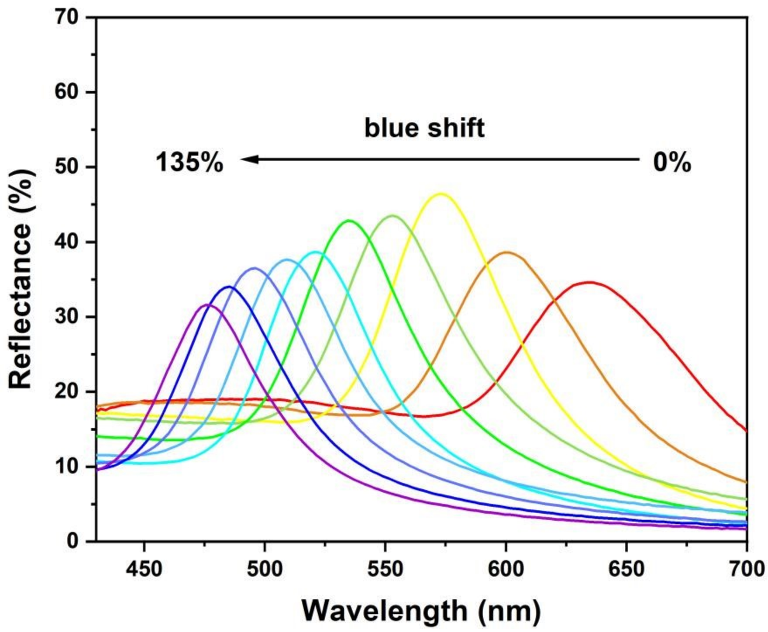
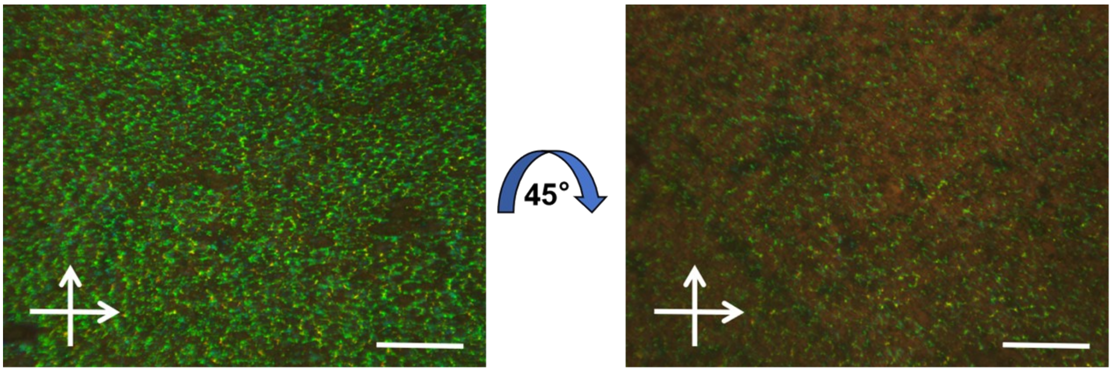


Figure S6. UV-vis reflection spectra of B-CLCE at room temperature with strains of 0%, 15%, 30%, 45%, 60%, 75%, 90%, 105%, 120% and 135%.



10

Figure S7. POM images of B-CLCE with strain of 60%. The sample is rotated by 45°. Scale bar: 200 μm.

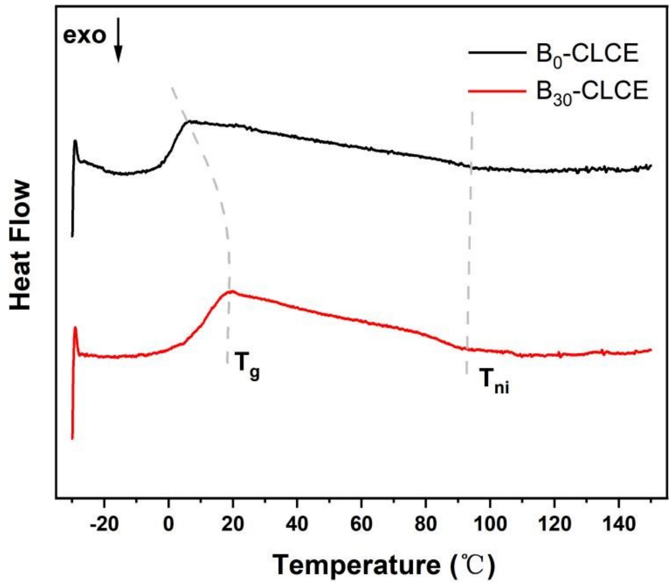


Figure S8. DSC curves of B_0-CLCE and B_{30}-CLCE during the second heating cycle.

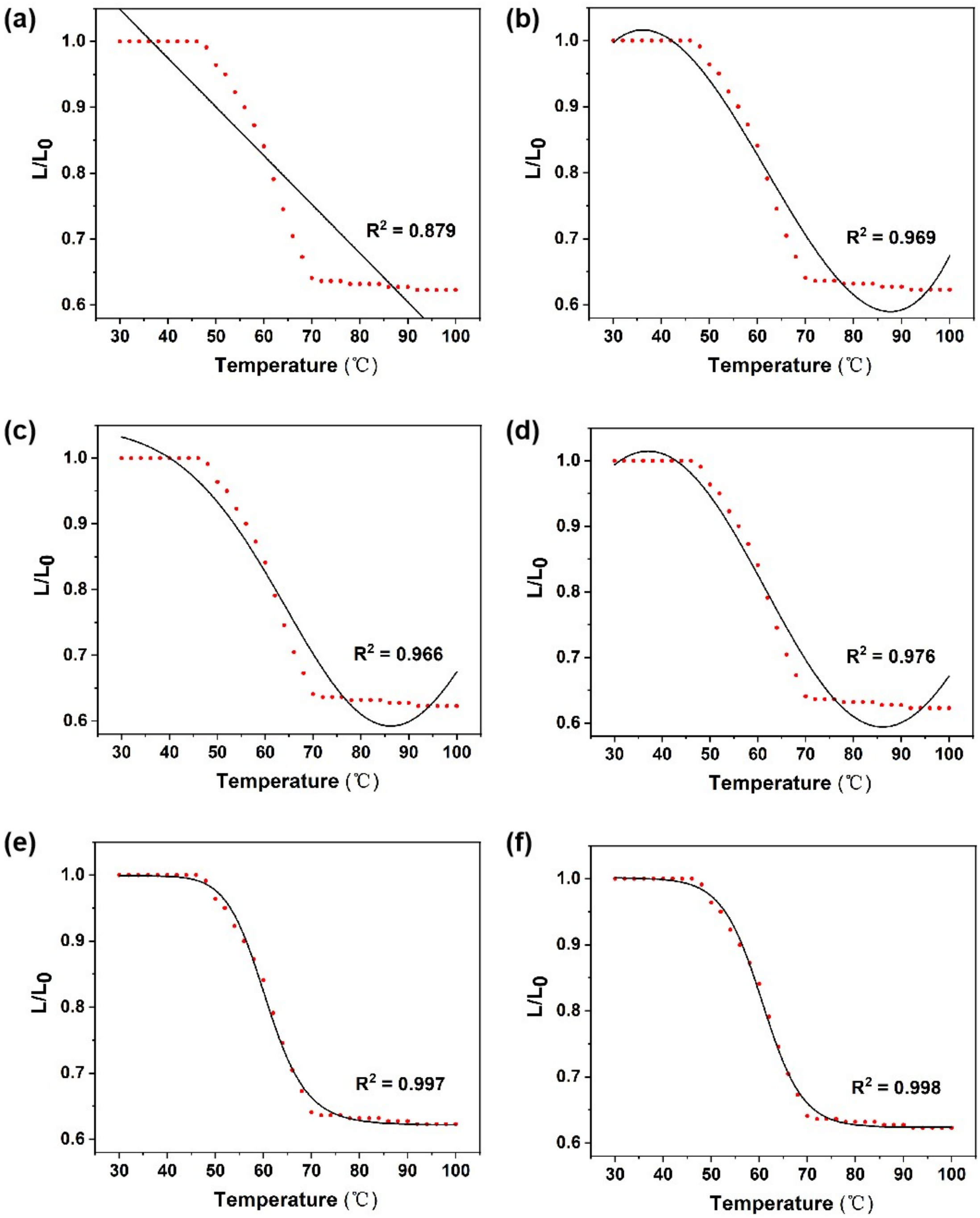


Figure S9. The fitting results using the (a) Linear, (b) Cubic, (c) Gauss, (d) Sine, (e) Logistic and (f) Boltzmann function.

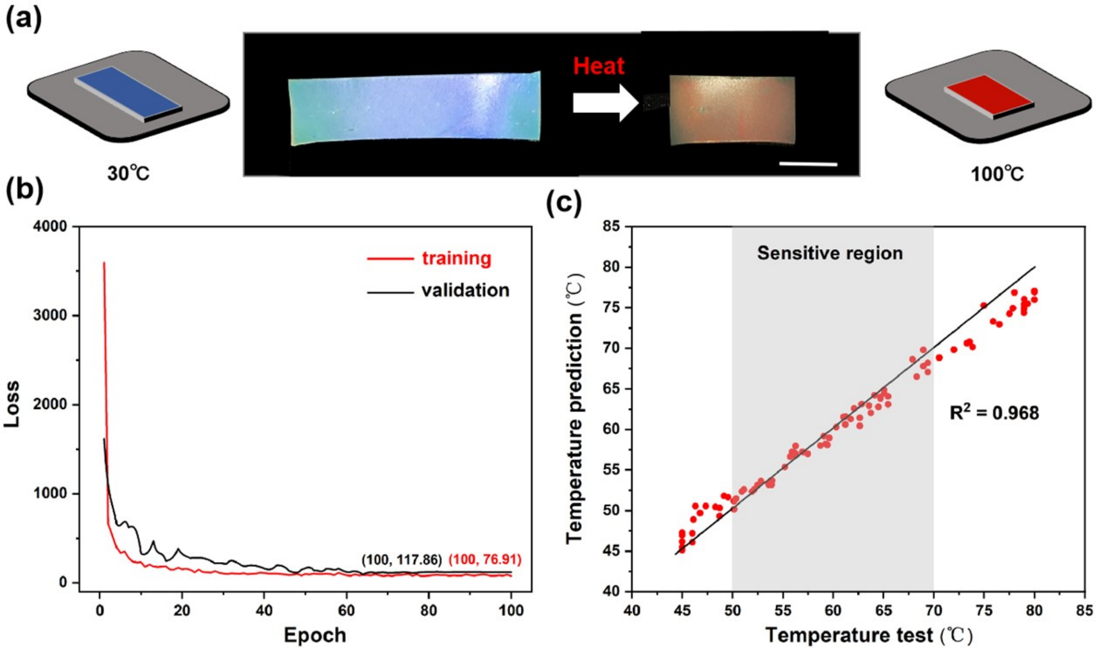


Figure S10. (a) Schematic diagram of the heating-induced color change of the blue-reflective B-CLCE, (b) loss curve of the CTVPM training process, and (c) scatter plot of the temperature prediction results (about 100 data points). Scale bar: 5 mm.

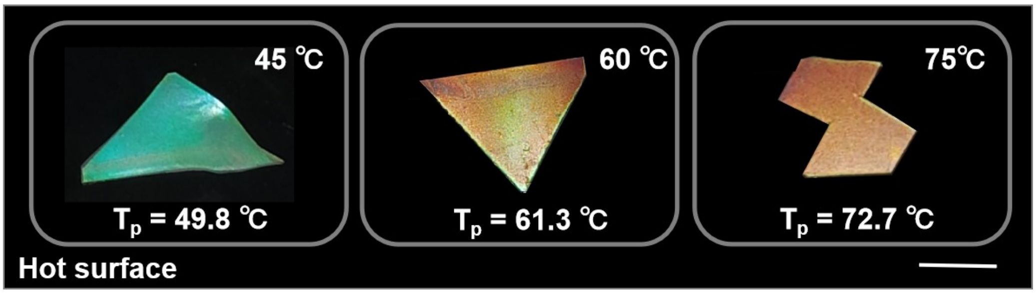


Figure S11. Temperature prediction results of the Color-based Temperature Value Prediction Model (CTVPM) for B-CLCEs of different shapes. T_p: predicted temperature. Scale bar: 5 mm.

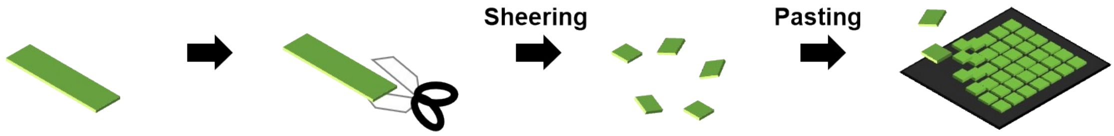


Figure S12. Flowchart for the preparation of B-CLCE array using dynamically bond-induced oriented B-CLCE.

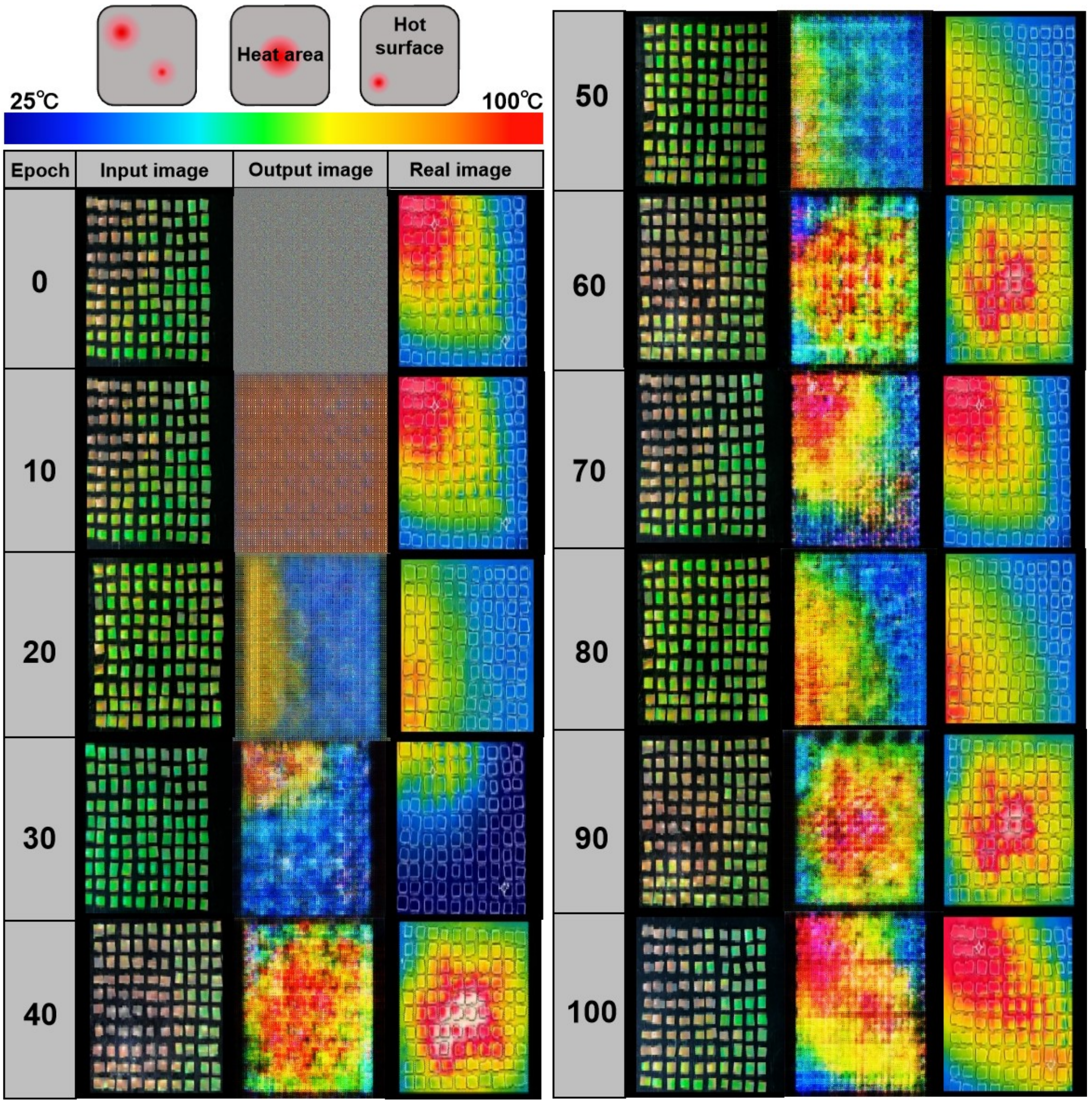


Figure S13. Input images, output images, and real images corresponding to different epochs during the deep learning.

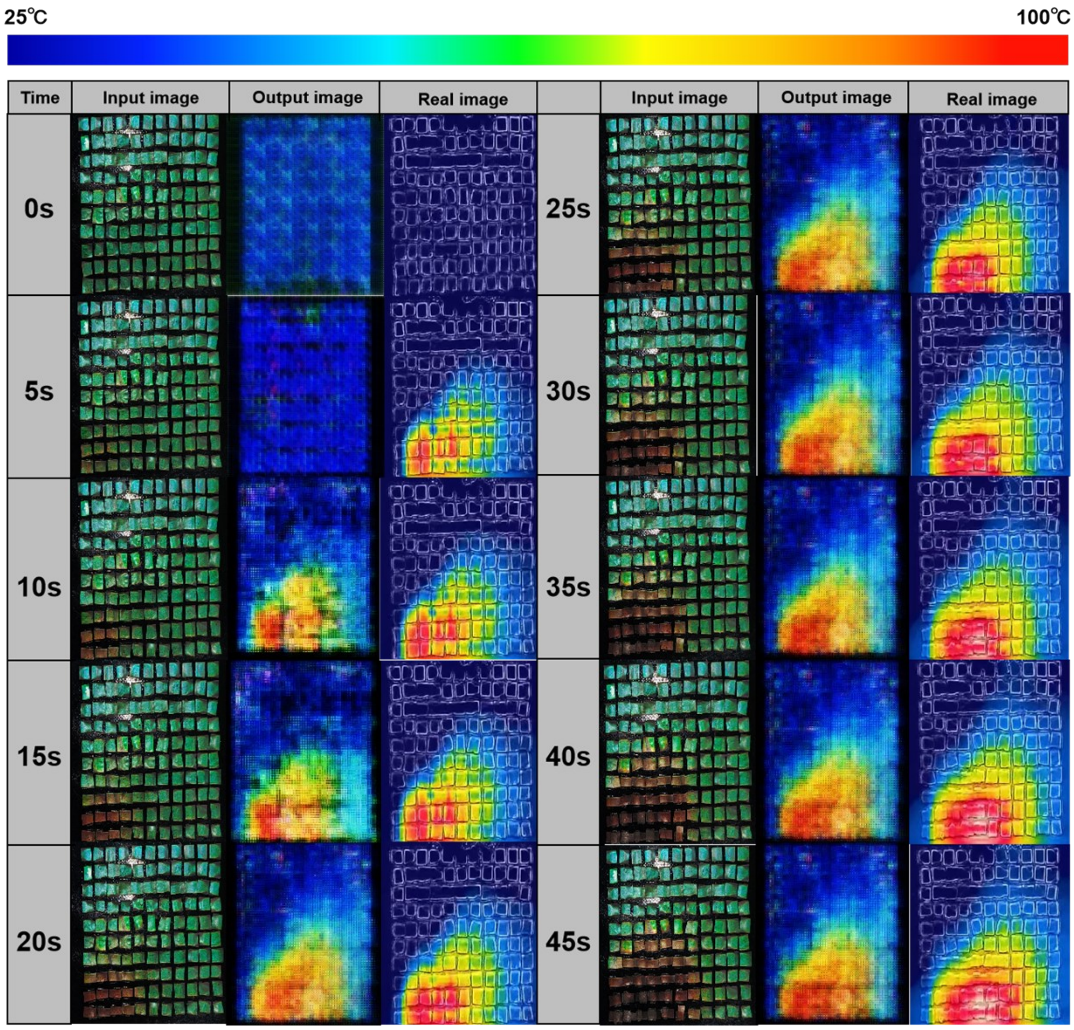


Figure S14. Local heating with a lighter flame at the bottom-left position of the B-CLCE. The results of capturing the optical images of the B-CLCE array every 5 seconds and performing temperature maps identification and mapping through CATMM. Heating with a lighter flame at the bottom-left position.

Table S1. The layer configuration of deep learning network for Color-based Temperature Value Prediction Mmodel (CTVPM).

Layer no.	Layer type	Filters	Kernel Size / Strides	Output Dimensions
0	Input layer	-	-	(554, 960, 3)
1	Conv2D + BN + ReLU	32	3 × 3 / same	(554, 960, 3)
2	Conv2D + BN + ReLU	32	3 × 3 / same	(554, 960, 3)
3	MaxPool + Dropout (0.3)	-	2 × 2 / 2	(277, 480, 32)
4	Conv2D + BN + ReLU	64	3 × 3 / same	(277, 480, 32)
5	Conv2D + BN + ReLU	64	3 × 3 / same	(277, 480, 32)
6	MaxPool + Dropout (0.3)	-	2 × 2 / 2	(139, 240, 64)
7	Conv2D + BN + ReLU	128	3 × 3 / same	(139, 240, 64)
8	Conv2D + BN + ReLU	128	3 × 3 / same	(139, 240, 64)
9	Conv2D + BN + ReLU	128	3 × 3 / same	(139, 240, 64)
10	MaxPool + Dropout (0.3)	-	2 × 2 / 2	(70, 120, 128)
11	Conv2D + BN + ReLU	256	3 × 3 / same	(70, 120, 128)
12	Conv2D + BN + ReLU	256	3 × 3 / same	(70, 120, 128)
13	Conv2D + BN + ReLU	256	3 × 3 / same	(70, 120, 128)
14	MaxPool + Dropout (0.3)	-	2 × 2 / 2	(35, 60, 256)
15	Conv2D + BN + ReLU	256	3 × 3 / same	(35, 60, 256)
16	Conv2D + BN + ReLU	256	3 × 3 / same	(35, 60, 256)
17	Conv2D + BN + ReLU	256	3 × 3 / same	(35, 60, 256)
18	MaxPool + Dropout (0.3)	-	2 × 2 / 2	(17, 30, 256)
19	Flatten	-	-	(138240)
20	Dense + ReLU + Dropout (0.3)	256	-	256
21	Dense + ReLU + Dropout (0.3)	128	-	128
22	Dense	1	-	1

Table S2. The layer configuration of generator model.

Layer no.	Layer type	Filters	Kernel Size / Strides	Output Dimensions
0	Input layer	-	-	(224, 192, 3)
1	Conv2D + SeLU	32	3 × 3 / 2	(112, 96, 32)
2	MaxPool	-	2 × 2 / 2	(56, 48, 32)
3	Conv2D + SeLU	64	3 × 3 / 2	(28, 24, 64)
4	MaxPool	-	2 × 2 / 2	(14, 12, 64)
5	Conv2D + SeLU	128	3 × 3 / 2	(7, 6, 128)
6	Trans Conv2D + SeLU	128	3 × 3 / 2	(14, 12, 128)
7	Trans Conv2D + SeLU	64	3 × 3 / 2	(28, 24, 64)
8	Trans Conv2D + SeLU	32	3 × 3 / 2	(56, 48, 32)
9	Trans Conv2D + SeLU	16	3 × 3 / 2	(112, 96, 16)
10	Trans Conv2D + Tanh	3	3 × 3 / 2	(224, 192, 3)
11	Reshape	-	-	(224, 192, 3)

Table S3. The layer configuration of discriminator model.

Layer no.	Layer type	Filters	Kernel Size / Strides	Output Dimensions
0	Input layer	-	-	(224, 192, 3)
1	Conv2D + SeLU	128	3 × 3 / 2	(112, 96, 128)
2	MaxPool	-	2 × 2 / 2	(56, 48, 128)
3	Conv2D + SeLU	64	3 × 3 / 2	(28, 24, 64)
4	MaxPool	-	2 × 2 / 2	(14, 12, 64)
5	Conv2D + SeLU	32	3 × 3 / 2	(7, 6, 32)
6	Flatten + Dropout (0.4)	-	-	(1344)
7	Dense + SeLU + Dropout (0.4)	512	-	512
8	Dense + SeLU + Dropout (0.4)	64	-	64
9	Dense + Sigmoid	1	-	1

Supporting Movie

Movie S1. Optical (top) and infrared (bottom) images of the heating process in one local heating source region (2×speed). Scale bar: 1cm.